Experiments with Japanese Text Retrieval using *mg*

Phil Vines
Department of Computer Science
RMIT
phil@cs.rmit.edu.au *Ross Wilkinson*
CSIRO
ross.wilkinson@cmis.csiro.au

August 1999

Abstract

We used the mg *syetem to experiment with automated indexing and retrieval of Japanese documents. We tried several different indexing strategies as well as combination of evidence techniques.*

Keywords Japanese Information Retrieval

1 Introduction

This paper documents work carried out at RMIT and CSIRO on the NACSIS Japanese text collection. We experimented with ad-hoc queries, using fully automatic querying and retrieval. In the run which we submitted for evaluation, shown in Table 2, we obtained an average (non interpolated) precision of 0.212. This was obtained using standard character indexing. Since that time we have tried a number of different techniques, with our best result to date being an average precision of 0.359.

2 Japanese Text

Japanese text, like Chinese and Korean, has no spaces between words. Therefore any indexing technique which involves using words as index terms first requires a segmenter. We used the freely available *Chasen* [3] system for this purpose.

Unlike Chinese text which consists entirely of one type of characters — called *Hanzi* in Chinese, or Kanji in Japanese — Japanese also contains Katakana and Hiragana, as well as some romanized text. The *Katakana* characters are typically used for words borrowed from other languages, especially English. Each character usually represents a consonant plus vowel combination. The *Hiragana* characters are typically grammatical function words, again, each character representing a syllable. When had hoped to experiment with the text type as one of the indexing properties, but have not done so to date.

3 Collection Characteristics

Some statistics relating to the size and numbers of distinct terms according to the indexing method used are shown in Table 1

4 The Retrieval System

We used the *mg* [4] system for our retrieval experiments. It implements a vector space document query similarity model, and supports a variety of similarity measures. The *mg* system is designed to handle

Table 1: *Collection statistics for the NTCIR Japanese collections.*

	Size (Mb)	Number of documents	Distinct characters	Distinct words	Distinct bigrams
Japanese coll'n	312.7	332,931	8,457	267,923	837,805

English text, and so some pre processing is required in order to handle Japanese. For character based retrieval, we converted each 16 bit Japanese character to a four ASCII character string by using a hexadecimal representation.

5 Experiments

5.1 Character Indexing

We experimented with ad-hoc queries, using fully automatic querying and retrieval. In the run which we submitted for evaluation, shown in Table 2, we obtained an average (non interpolated) precision of 0.212. This was obtained using standard character indexing. Following an examination of the dictionary built for the collection, we noticed that there was a large number of English words in the dictionary, many of which only occurred once. We then pre-processed the Japanese text, removing all the English terms, prior to the indexing of the collection. This produced a notable improvement in the recall-precision performance. The results of this are shown in Table 3. As yet we have not had time to fully investigate the reasons for this. We speculate that it is to do with the change in *tf.idf* weightings, as there are 8,457 index terms when English text is excluded, and 546,246 index terms when the English text is included.

5.2 Word Indexing

We applied the *Chasen* morphological analyzer to segment both the documents and queries into words. With the English text removed we obtained an average interpolated precision of 0.312. The results of this run for the test queries are shown in table 4.

5.3 Bigram Indexing

Bigram indexing has proven to be successful in Chinese retrieval [2], and so seemed an obvious technique to experiment with. The bigrams we used were simply all pairs of adjacent Japanese characters. So for characters $ABCD$, we generated pairs AB, BC, CD. One of the major practical problems with bigram indexing is the size of the index created. The bigram database without English words contained 837,805 terms, while the one with English terms contained 1,379,443 terms. In the processing of the bigrams where the English text was retained, it was simply passed through the preprocessor unaltered. Thus the string $AB < Eng > CD$, where $< Eng >$ is an English word would produce AB, $< Eng >$, BC, CD.

The average non interpolated precision of all queries was 0.332 for the bigrams which included English text. For bigrams without the English text, the average precision was about 0.347, or about 3 percentage points better than for word indexing. More details of the results are shown in table 5.

5.4 Combination of Evidence

Previous work that we have done on a Chinese text collection [1] has shown that combination of evidence from different sources tends to produce slightly improved results. We experimented with our two best approaches thus far – words without English, and Bigrams without English. The combination technique we used involved normalising the similarity values returned by each measure, then calculating a combined value using simple linear combination, i.e. $sim_{new} = 0.5 * sim_1 + 0.5 * sim_2$. We then reranked the documents based on the new similarity measure. Obviously documents that are returned by both methods will be prefered over documents which

Table 2: Retrieval Performance using Character based indexing, with English

Char Indexing	0%	20%	40%	60%	80%	100%	Av precision
	0.601	0.365	0.247	0.137	0.067	0.0286	0.212

Table 3: Retrieval Performance using Character based indexing - English removed

Char Indexing	0%	20%	40%	60%	80%	100%	Av precision
No English	0.687	0.455	0.293	0.189	0.098	0.039	0.26395

were only returned by one of the methods. This produced a further 1.2 percentage point average improvement. Results are shown in Table 6.

6 Conclusion

We experimented with indexing using character, word and bigram indexing, as well as simple combination of evidence. Of the three baseline indexing techniques investigated, bigram indexing gave the best performance. A simple combination of the word and bigram retrieval methods gave a slightly improved performance. There where a number of other modifications which we would have liked to have tried, but time did not permit.

References

[1] M. Fuller, M. Kaszkiel, C. L. Ng, P. Vines, R. Wilkinson, and J. Zobel. MDS trec-6 report. In *Proc. Text Retrieval Conference (TREC)*, pages 241–258, 1997.

[2] K. L. Kwok. Comparing representations in chinese information retrieval. In *Proc. ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 34–41, Philadelphia, July 1997.

[3] Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Osamu Imaichi, and Tomoaki Imamura. Japanese morphological analysis system chasen. Available from http://cactus.aist-nara.ac.jp/lab/nlt/chasen/distribution.html.

[4] I.H. Witten, A. Moffat, and T.C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Van Nostrand Reinhold, New York, 1994.

Table 4: Retrieval Performance using Word based indexing

Word Indexing	0%	20%	40%	60%	80%	100%	Av precision
No English	0.746	0.512	0.385	0.248	0.111	0.03 4	0.312

Table 5: Retrieval Performance using Bigram based indexing

Bigram Indexing	0%	20%	40%	60%	80%	100%	Av precision
English	0.730	0.518	0.409	0.275	0.152	0.059	0.332
No English	0.754	0.533	0.424	0.292	0.156	0.055	0.347

Table 6: Combination of Evidence Using Words and Bigrams

Combination of Evidence	0%	20%	40%	60%	80%	100%	Av precision
	0.791	0.556	0.455	0.306	0.147	0.053	0.359